Computer Systems Science & Engineering
DOI:10.32604/csse.2021.014975
Review

A Review of Dynamic Resource Management in Cloud Computing Environments

Mohammad Aldossary*

Department of Computer Science, College of Arts and Science, Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia
*Corresponding Author: Mohammad Aldossary. Email: mm.aldossary@psau.edu.sa
Received: 30 October 2020; Accepted: 27 November 2020

Abstract: In a cloud environment, Virtual Machines (VMs) consolidation and resource provisioning are used to address the issues of workload fluctuations. VM consolidation aims to move the VMs from one host to another in order to reduce the number of active hosts and save power. Whereas resource provisioning attempts to provide additional resource capacity to the VMs as needed in order to meet Quality of Service (QoS) requirements. However, these techniques have a set of limitations in terms of the additional costs related to migration and scaling time, and energy overhead that need further consideration. Therefore, this paper presents a comprehensive literature review on the subject of dynamic resource management (i.e., VMs consolidation and resource provisioning) in cloud computing environments, along with an overall discussion of the closely related works. The outcomes of this research can be used to enhance the development of predictive resource management techniques, by considering the awareness of performance variation, energy consumption and cost to efficiently manage the cloud resources.

Keywords: Cloud computing; resource management; VM consolidation; live migration; resource provisioning; auto-scaling

1 Introduction

Cloud computing has changed the way in which the businesses and individuals are used the Information Technology (IT) by offering their customers on-demand services such as applications, platforms and infrastructures at competitive prices depending on their usage (e.g., *pay-as-you-go model*). However, the widespread adoption of cloud computing and the rising number of cloud customers have increased the overall operating costs for cloud providers [1–5]. Thus, reducing the operational costs of different cloud services is an active area of research.

A number of mechanisms have been adopted by cloud service providers in order to achieve economies of scale in a cloud environment [6]. For example, dynamic consolidation presents a solution to improve resource utilization and achieve energy efficiency in clouds. Virtual Machines (VMs) consolidation allows VMs to move from one Physical Machine (PM) to another through live migration, without any interruption to the service. This mechanism plays a major role in load balancing between the PMs and

reduces the overall energy consumption by switching off the idle hosts. However, live migration of the VMs is a resource-intensive operation that affects the performance of the migrating VM and therefore the services running on other VMs [7]. Also, there are additional costs [8] in terms of migration time and energy overhead, which need to be explored further [9]. Therefore, understanding the impact of VM live migration is essential to design an efficient VM consolidation strategy. Resource provision defined as VMs auto-scaling is another solution to provide additional capacity to the VMs on-the-fly in order to handle service performance variations. However, it can take a few minutes for this process to start [10], which is inappropriate for VMs that need to scale rapidly during computation [11]. In fact, there are additional costs [8] in terms of scaling time (booting/rebooting), license fees for the new VMs (horizontal scaling) and energy overhead that need attention [12]. Hence, understanding the impact of VMs auto-scaling is important to design an efficient resource provision technique.

Furthermore, most of the literature studies have concentrated on reducing energy consumption and optimizing resource utilization, rather than enhancing service performance. To illustrate that, cloud providers such as Amazon EC2 [13] have developed their Service Level Agreements (SLAs) based on the availability of services, without such a service performance assurance [14]. For example, consider the situation where a number of VMs run on the same PM, and each VM is allocated its fair share of resources. If the workload of the VM's increases and no resources are sufficient to manage this increasement (e.g., the workload reaches the upper level of Central Processing Unit (CPU) such as 95% threshold). In this case, there may be resource competition leading to VMs' performance degradation, which may affect the fulfilment of the SLAs and therefore the revenue of the cloud service provider. Thus, *predictive* mechanisms have the advantage of taking preventive actions (e.g., live migration and auto-scaling) at an early stage to avoid service performance degradation.

The aim of this research is to investigate the dynamic resource management issues and the impact of VMs consolidation and resource provisioning in cloud computing environments. This would help to enhance the development of *predictive* resource management techniques, by considering the awareness of performance variation, energy consumption and cost to efficiently manage the cloud resources.

The remainder of this paper is organized as follows: Section 2 presents the fundamental concepts of cloud computing with a description of its definition, services types, deployment types and virtualization technologies. The aspects of cloud applications and their workload patterns as well as related benchmarks are discussed in Section 3. Section 4 reviews the existing work on cloud resource management, including VMs consolidation and resource provisioning. Section 5 includes the overall discussion, along with a comparison summary of the closely related works. Section 6 concludes this paper.

2 Overview

2.1 Cloud Computing

Cloud computing is a technology that uses the internet to provide computing resources as services. This innovation allows scalable, on-demand sharing of resources and their costs between cloud customers. Also, it provides customers with various online computing services at reasonable prices, to manage, process, and store their data efficiently. With the cloud, customers do not need to install any kind of software on their machines; as long as the internet connection is accessible, they can reach their data worldwide from any computer [15].

The cloud computing system architecture consists of three standard layers, *Software-as-a-Service (SaaS)* where the service is developed, *Platform-as-a-Service (PaaS)* where the service is deployed, and *Infrastructure-as-a-Service (IaaS)* where the service is run [16]. Furthermore, cloud computing can be deployed through many models, which can be mainly *Public*, *Private*, *Hybrid*, and *Community* clouds [17].

2.2 Virtualization

Virtualization is a key component of the cloud computing infrastructure and is defined as: "*a technology that combines or divides computing resources to present one or many operating environments using methodologies like hardware and software partitioning or aggregation, partial or complete machine simulation, emulation, time-sharing, and many others*" p.2, [18]. One of the main advantages of virtualization is to abstract the Physical Machines (PMs) hardware in order to provide Virtualized Machines (VMs) that can work in isolation and run different applications with different operating systems. By virtualization, the VMs can be consolidated to minimize the number of active PMs using (e.g., live migration), which would then reduce the power consumption as well as lowering the operational cost. Thus, virtualization adds an essential value to the cloud infrastructure by increasing the physical resource utilization, achieving significant energy savings and reducing the operational cost in cloud environments [19].

2.3 Virtual Infrastructure Manager

Cloud infrastructure providers use Virtual Infrastructure Manager (VIM) to manage their physical resources in order to provide virtualized resources to meet their customers' service requirements. In order to build, deploy and manage cloud infrastructures, there are several open-source cloud management platforms available to manage virtualized infrastructures in clouds. Some examples of the major open source cloud platforms are OpenNebula [20], OpenStack [21] and CloudStack [22]. The following Tab. 1 summarizes some of the features of these VIMs.

Table 1: Comparison of open-source cloud platforms

Functionality	OpenNebula	OpenStack	CloudStack
Cloud infrastructure	Private, Public and Hybrid Clouds	Private, Public and Hybrid Clouds	Private, Public and Hybrid Clouds
Resource abstraction	Compute, Storage and Network	Compute, Storage and Network	Compute, Storage and Network
Architecture	Modular (third- party component)	Fragmented into many modules	Monolithic central controller
Installation difficulty	Easy (process-based package installers)	Difficult (many choices, not fully automation)	Medium (Few parts to install)
Supported hypervisors	Xen, KVM, VMWare, vCenter	Xen, KVM, VMware, HyperV, vCenter, LXC, vSphere	Xen, KVM, VMWare, HyperV, LXC, vSphere,
Administration	Web UI, CLI	Web UI, CLI	Web UI, CLI
User management	Yes	Yes	Yes
Live migration	Yes	Yes	Yes
Load balancing	Yes	Yes	Yes
Fault-tolerance	VM scheduling, replication	VM scheduling, replication	VM scheduling, replication
High availability	Yes	Yes	Yes
Security	User authentication	VPNs, firewall, user authentication, others	VPNs, firewall, user management, others
Compatibility	All Amazon interfaces	Amazon EC2, Amazon S3	Amazon EC2, Amazon S3
Extensibility	Yes	Yes	Yes

OpenNebula, OpenStack and CloudStack have a common role in providing a platform for deploying, managing and provisioning (compute, storage and networking) resources through interfaces such as Web User Interface (Web UI) and Command Line Interface (CLI). However, there are some differences in terms of their architectures based on the configurations, settings and their deployment. For instance, OpenStack has many components to install, which may increase the complexity of installation and configuration as well as the management overhead [23]. In order to avoid this, the OpenStack administrator has to only install the required components to meet the needs of their cloud deployment. In contrast, OpenNebula does not have such constraints as it provides centralized deployment and has a fine-grained core [23]. In addition to OpenNebula, OpenStack and CloudStack, there are other VIMs available freely or commercially for the deployment and management of cloud infrastructures such as OpenQRM [24], Eucalyptus [25], Nimbus [26] and others more.

2.4 Hypervisors

Hypervisors-based virtualization abstracts the underlying physical hardware to provide isolated instances, called VMs, which can run their own operating system (guest-OS) [27]. These VMs are managed by the hypervisor, which is also referred to the Virtual Machine Monitor or Manager (VMM) to control the number of resources allocated to each VM. The hypervisor sits between the physical hardware and OS, which is also responsible for creating, running, migrating, copying, and deleting the VMs [27]. Further, hypervisors can be implemented in different ways such as full virtualization when the hypervisor runs on underlying physical OS and hardware virtualization when the hypervisor runs on underlying physical hardware. Some examples of hypervisors include Kernel-based Virtual Machine (KVM) [28], Xen [29], VMware [30] and Virtual Box [31].

2.5 Containers

Containers-based virtualization modifies the underlying host OS to provide isolated instances, called *containers*, that can run different applications by sharing the same host OS [27]. Containers provide new ways for faster-running applications, developing, and shipping. It represents a light-weight alternative instance when compared to VM, thus, instead of building one application, developers can build a suite of components, called *micro-services*, which come together over the container [32]. Most of cloud service providers have moved to *Docker* [33] such as Microsoft, Google and Amazon Web Services to provide the infrastructure that supports the container standard [34]. Containers are better suited to micro-services than VMs, they can start up and shut down more rapidly as well as their resources can be scaled independently. However, containers do not provide full isolation, which may cause security issues. Therefore, hypervisor-based is more appropriate than container-based virtualization in terms of isolation and security concern. Some examples of containers include Docker [33], Linux Containers (LXC) [35] and Warden Container [36].

3 Cloud Computing Applications

Cloud applications should be designed specifically with the support of a cloud computing architecture; thus, the applications need to break down into separate components to support the distribution among cloud resources. Also, the cloud applications should be designed to support scalability and elasticity, which allow dynamic reservation and release of the cloud resources to match the changes of the workloads.

3.1 Workload Patterns

In cloud environments, different applications have different resource usage requirements. Cloud applications may also experience different patterns of workload depending on the customers' usage behaviors, and these patterns of workload consume power differently based on the services and resources

they use. As indicated in Fehling et al. [37], the cloud workload patterns can be categorized as *static workload, periodic workload, once-in-a-lifetime workload, unpredictable workload*, and *continuously changing workload*.

As depicted in Fig. 1, *a static workload pattern* occurs when an application is running continuously with the same and stable resource utilization over a period of time. Private websites and wikis are examples of such static workload. A *periodic workload pattern* can be experienced when an application is running with a repeated resource utilization peaks occurring over time intervals (e.g., seasonal changes). Examples of this type of workload include shopping websites during holiday periods, sporting events (Olympics) and traffic during rush hours.

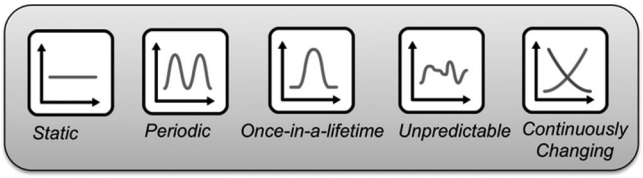

Figure 1: Cloud application workload patterns [37]

Furthermore, when an application is running with stable resource utilization and peak once over time, it is considered *once-in-a-lifetime workload pattern*. Payroll, billing and backup applications are examples of once-in-a-lifetime tasks or jobs. An *unpredicted workload pattern* occurs when an application has a random peak (constantly fluctuating) of resource utilization over time. Unpredictable traffic and forecasting are examples of unpredicted workload. Finally, when the application is running with stable resource utilization and rapidly decreases or increases over time, it experiences a *continuously changing workload pattern* [37]. Examples of such type of workload include social networking (Facebook and Twitter), open-source downloads and Android applications.

As mentioned early, these types of application workload patterns can have a different impact on energy consumption based on the resources they consume.

3.2 Benchmarking

Benchmark suites are adopted to evaluate cloud services to support the configuration and adaptation of applications before they start utilizing cloud resources, such as VMs and containers. Benchmarking aims at defining and reproducing execution conditions for the target system (application, resource, service) to be evaluated [38]. It also provides a set of metrics in order to quantify the relative software and hardware performance, and understand how cloud application workloads behave as the underlying cloud resources are stretched and approach full capacity [39].

In this regard, the Standard Performance Evaluation Corporation (SPEC) [40] launched a tool that provides a set of synthetic workloads, which exercises the CPU, memory and disk performance as well as tests the energy efficiency of a system at different load levels. Generally, this benchmark exerts graduated levels of load on a given machine, normally evaluating the energy consumption and performance of server hardware between (idle 0% and fully active 100%) load at 10% graduated load levels.

Similarly, a simple benchmarking tool for POSIX systems, called *Stress-ng* [41], has been designed as a workload generator. This tool has the capability to simulate a wide range of workload patterns such as static, periodic, continuously changing, and once-in-a-lifetime workload patterns. Further, the *Stress-ng* workload generator is able to simulate both single and multi-threaded applications, as well as test workloads that are resource-bound in many ways, e.g., applications that are both CPU and memory intensive.

4 Dynamic Resource Management in Cloud Computing

Resource management is one of the most important problems in cloud infrastructures, which can be expressed as a multi-objective problem since there are several conflicting objectives (e.g., maintain the performance, reduce energy and costs) that need to be optimized [9,42]. Therefore, cloud service providers have applied dynamic resource management through VMs' consolidation and resource provisioning techniques in order to meet the performance requirements of applications, while minimizing the operation costs and energy consumptions in cloud data centers.

4.1 VM Consolidation

One of the benefits of virtualization is the VMs' consolidation strategy, which allows cloud service providers to migrate and reallocate the VMs from one host to another in order to increase resource utilization and reduce energy costs in cloud data centers. The aggregation of VMs through live migration, therefore, has a significant impact on energy efficiency by gathering several VMs into the minimum number of hosts and switching the idle hosts into a power-saving mode. However, VM consolidation is not a trivial task in case of unpredicted increases in demand, as it can result in generates unnecessary migrations, violations of the SLA and increases the operation cost due to the migration processes [43]. Therefore, dynamic VMs consolidation requires an estimate of the workload demand in order to handle the fluctuating demands of cloud customers, efficiently manage cloud resources and avoid unnecessary migrations [44].

VM live migration acts as a backbone of the VM consolidation process, which can be defined as the capability of transferring a complete state of the VM (including CPU states, memory pages, storage and network connections) from the source host to the destination host, without any interruption in the service or application [45,46]. There are two types of VM migration, which are currently used in cloud data centers, namely, *post-copy* and *pre-copy* migration.

- ***Post-copy***: Transfers a VM's memory contents after its processor state has been sent to the destination host. However, this method can take a long migration time, which consumes the resources on both source and destination hosts due to the residual dependency. Also, it has some downtime initially, which makes the VM's service unavailable for a certain time period [47].
- ***Pre-copy***: First copies the memory state to the destination, through iterative phases, after which its processor state is transferred to the destination. In this way, the VM can be migrated from one host to another with a close to zero downtime [48].

Live migration efficiency of multiple VMs has been investigated in various research studies. For instance, Ye et al. [45] presented a live migration framework of multiple VMs based on different resource reservation mechanisms. This framework aims to improve migration efficiency by using parallel migration and workload-aware migration strategies. Experimental results show that the performance overheads of the live migration process are affected by workload types, memory size and the number of CPUs. Thus, parallel migration and workload-aware migration strategies can efficiently improve the performance of migrated VMs. However, the performance overhead incurred by concurrent VM migrations may increase the migration interference on the destination host.

Zhao et al. [49] presented a VM placement method based on VM service performance, which aims to address VMs performance degradation issue when placing the VMs. This method takes the application-aware resource consumption characteristic into consideration to place the VMs on appropriate PMs in order to guarantee the VM performances and ensure customers' Quality of Experience (QoE). The proposed method is evaluated in a real cloud platform (OpenStack) using video streaming applications. The results show that the proposed method can minimize PM performance degradation and guarantee the VM performance compared to other methods. However, their approach only focuses on the resource

consumption characteristic when performing VMs placement and does it not take the power consumption of the PMs and VMs into account.

Moreover, Ferreto et al. [50] proposed an approach called dynamic consolidation with migration control, which aims to reduce the number of VM migrations and the number of active hosts using linear programming formulation. This approach gives a higher priority to migrate VMs with variable workload instead of the VMs with a stable workload in order to reduce the number of migrations and required hosts with a minimal SLA violation. They compared the proposed approach with static and dynamic consolidation approaches using TU-Berlin and Google data center workloads. The evaluation results demonstrate that the suggested approach performs well in terms of the number of PMs used and VMs migrated. However, this approach does not take into account VMs power consumption and migration costs when consolidating the VMs.

Farahnakian et al. [46] presented a modified approach of Best Fit Decreasing (BFD) algorithm, named a Utilization Prediction-aware Best Fit Decreasing (UP-BFD) algorithm. This approach employed a utilization prediction model to eliminate unnecessary VM migrations and reduce SLA violations using K-Nearest Neighbor Regression (K-NNR) model. The prediction model is trained by generating historical data based on different types of workloads developed in the CloudSim. This approach also considers both the current and future utilization of resources in order to perform VM consolidation based on the hosts CPU and memory utilization thresholds. Although this work focuses on reducing PMs energy consumption, the number of VM migrations and SLA violations, they do not consider the impact of energy consumption that occurs by VMs live migration decisions in their approach.

Further, Beloglazov et al. [51] addressed the problem of VMs consolidations under Quality of Service (QoS) constraints in cloud data centers. They employed the Markov chain model and the control algorithm to detect the overloaded hosts and then migrate some VMs in order to achieve a specified QoS goal. This dynamic VMs consolidation aims to improve the PMs resource utilization (particularly CPU utilization) for stationary workloads, which also can be applied for non-stationary workloads using the Multisize Sliding Window workload estimation technique. Simulation results using workload traces on PlanetLab servers demonstrate that the introduced method outperforms the benchmark methods while meeting the QoS goal. However, this method focused on improving the performance of cloud applications by reducing the number of overloaded hosts, but without explicitly considering energy and cost of VMs migrations, as a part of VMs consolidation decision criterion.

Xu et al. [52] proposed a lightweight interference-aware VM live migration strategy, called iAware. It focuses on the performance of VMs during and after live migration, considering the interference of the migration process on both source and destination PMs. The iAware jointly estimates, analyses and minimizes both the migration time and co-location interference among VM's based on a multi-resource demand and supply estimation model. The experiments are conducted in a real cloud environment with different workloads using a Xen hypervisor cluster platform. The results are compared with traditional interference-unaware algorithms and show that the iAware can estimate VM performance interference during live migration and meet the SLA requirements. However, their work does not consider the energy consumption overhead of VMs migrations.

Beloglazov et al. [53] presented an energy efficient resource management policy for cloud data centers. The proposed method mainly focuses on dynamic re-allocation of VMs using live migration in order to minimize the energy consumption, while maintaining the QoS requirements. They evaluated the proposed method using a CloudSim and the results show a reduction of energy consumption in a cloud data center. However, the proposed method does not show the effectiveness of the heterogeneity of the PMs in terms of energy efficient when performing the live migration of the VMs.

Furthermore, Beloglazov et al. [54] presented an energy-aware VM consolidation policies to optimize the resources utilization and energy efficiency in a cloud data center. In this approach, the VMs are migrated from one host to another in order to increase the overall servers' utilization and reduce infrastructure costs (energy costs) by switching off the idle hosts. Thus, upper and lower CPU utilization thresholds for each host are set along with several VM selection policies, in order to identify from which host the selected VMs should be migrated. The experiment results conducted in the CloudSim show that this approach leads to an improvement of energy efficiency in cloud data centers. Likewise, Farahnakian et al. [42] proposed a Self-Adaptive Resource Management System (SARMS) for efficient resource management in cloud infrastructure. The SARMS provides an adaptive utilization threshold (CPU and memory) mechanism to dynamically identify the overloaded and underloaded PMs. This system has two steps, migration of VMs from the overloaded PMs to prevent SLA violations, and consolidation of VMs into a minimum number of active PMs in order to reduce energy consumption. They evaluated the proposed system using the CloudSim based on real workloads from Google and PlanetLab. The obtained results show that the SARMS can achieve performance requirements, while reducing PMs energy consumption and the number of VM migrations. Nevertheless, these approaches do not consider the energy consumption overhead and the costs of VMs consolidation.

Beloglazov et al. [55] proposed a technique for dynamic VM consolidation based on CPU utilization thresholds. This technique focuses on cloud resource management strategies (e.g., VM migration) with the aim to optimize resource usage and reduce energy consumption, while maintaining the SLAs. It can be achieved by migrating the VMs from the underloaded hosts in order to reduce the number of active hosts and saving energy. To re-allocate the VMs, a Modified Best Fit Decreasing (MBFD) algorithm is used to sort the selected hosts based on their CPU utilization and energy efficiency. They evaluated the proposed technique through simulations with different types of workloads using PlanetLab servers. The results show that this technique outperforms other migration policies in terms of the number of VM migrations and SLA violation, while showing a similar level of energy consumption. However, the proposed technique lacks to consider the actual cost and power consumption caused by VMs consolidation.

Also, Malekloo et al. [56] introduced a Multi-objective Ant Colony Optimization (MACO) approach for VMs placement and consolidation algorithms. In this regard, the VMs' placement algorithm aims to minimize energy consumption, CPU resource wastage and communication cost. While, the VM consolidation algorithm aims to reduce SLA violations, VMs migration and the number of active PMs. They evaluated the proposed approach using the CloudSim based on eight performance metrics. The results show that this approach outperforms the other approaches in terms of achieving the balance between energy consumption, system performance and QoS requirements. Yet, this approach focused on minimizing PMs energy consumption without taking into consideration the energy consumption incurred by VMs consolidation.

Zhou et al. [57] proposed an adaptive strategy for energy and performance efficient VM consolidation, called (DADTA). The DADTA strategy aims to minimize energy consumption while satisfying the SLAs in the cloud data center. They applied a specific adjustment of thresholds to adapt the dynamic workload changes and then performed VM consolidation by using the DADTA in order to improve the overall optimization. To evaluate the proposed strategy, a modified prediction model conducted on the CloudSim is used to deal with the time-series data obtained from the Google cluster workload trace, and the findings show that the proposed DADTA outperforms other benchmarks in terms of minimizing the PMs energy consumption and SLA violations. In their work, the consolidated VMs are homogeneous and only considers PMs power consumption.

Moreover, Beloglazov et al. [43] presented adaptive algorithms for dynamic VM consolidation based on a statistical analysis of historical workload data. Statistical models are used to calculate the upper and lower

CPU utilization thresholds of each host. If the host is determined to be overloaded, one or more VMs are selected to be migrated from the host to another suitable one in order to optimize the resource usage and maintain a high level of SLAs. On the other hand, if the host is determined as underloaded, all hosted VMs are selected to be migrated from the host and switch it to the sleep mode in order to reduce the energy consumption. They evaluated the proposed algorithms through the CloudSim using workload traces from PlanetLab, considering the heterogeneity of PMs and VMs. The results of the experiments show that the proposed algorithms outperform other dynamic VM consolidation algorithms in terms of the level of SLA violations and the number of VM migrations. However, this work only considers PMs energy consumption and does not refer to VMs energy consumption.

Verma et al. [58] emphasized the importance of taking migration cost into account for a fine-grain VM consolidation strategy. Therefore, Zakarya et al. [59] proposed a VM consolidation technique, named a Consolidation with Migration Cost Recovery (CMCR). This technique aims to explore the ability of the VMs to recover their migration costs. In order to achieve that, the VMs should firstly be migrated to an energy efficient host and then continue to run them for a certain period of time. A linear power model is used to identify the power consumption for the target host in order to check the ability of the VMs to recover their migration costs. They evaluated the CMCR through CloudSim using real workload traces from a Google cluster. The results show that by using the CMCR the majority of the migrated VMs can recover their migration cost. However, their work is applicable only to the hosts that follow a linear power model and does not consider the heterogeneity of PMs or VMs. Similarly, Verma et al. [58] introduced a power-aware application placement framework for virtualized server clusters, called pMapper, which dynamically places the VMs to minimize the power consumption and the migration cost, while meeting the performance requirements. In their framework, they have extended the First Fit Decreasing (FFD) heuristic algorithm in order to migrate the VMs to suitable hosts. This is aimed to minimize the data center's energy consumption by reducing the number of active hosts, while taking into account the VMs migration cost. They have implemented the pMapper framework on IBM testbed with heterogeneous hosts using a set of benchmark applications. The results show that the pMapper outperforms other power unaware algorithms in terms of minimizing the PMs power consumption and VMs migration costs, while meeting the application performance guarantees. However, their framework does not provide any information regarding the migration costs calculation.

4.2 Resource Provisioning

Cloud service providers support an on-demand resource provisioning model, called auto-scaling, which provides additional resources requested by applications using vertical and horizontal scaling techniques. Generally, the auto-scaling can be defined as the ability of a system or users to add and remove resources (such as CPU, memory), which is beneficial for adapting to workload variations and ensuring consistent performance with lower costs [8,12]. Cloud providers such as Amazon Web Services (AWS) [60] offer this service.

Auto-scaling is a dynamic property for cloud computing, and it comes in two types, namely, *vertical* and *horizontal* scaling. The ***vertical scaling*** is used to add or release virtual resources dynamically (e.g., virtual CPUs and memory) inside the VMs, whereas ***horizontal scaling*** is used to create or delete VMs, all of which were based on application requirements. However, the latter mechanism may take a few minutes to initiate [10,61–63], which may be unsuitable for VMs that need to rapidly scale during the computation [11,64].

To achieve the scalability of cloud resources a combination of these two scaling techniques can help to find an optimal scaling strategy [63]. However, most of the vertical and horizontal scaling approaches are *reactive* methods which happen after detecting there are not enough resources for an application [64,65]. Thus, it is desirable if the methods can be scaled earlier than the time when the workload actually

increases. This can be achieved by using *proactive* methods that can predict workloads of applications and scale the resources commensurate with the predicted workload.

A number of solutions have been proposed to support resource elasticity for cloud applications. For example, Ficco et al. [9] presented a new approach for managing elastic resources reallocation in cloud infrastructures using the coral-reefs algorithm and game theory optimization. This approach uses a multi-objective optimization to maintain customers SLAs, minimize resource consumption and cost during the auto-scaling and migration processes. In their work, the coral-reefs algorithm is used to model the elasticity of cloud resources, whereas the game theory is used to optimize the aims of the service provider expressed through resource reallocation strategies with respect to the customer's requirements. The experimental results show that the combination of coral-reefs algorithm and game theory optimization achieves the elasticity of cloud resources and leads to significant performance improvements. However, the energy-related cost when performing the auto-scaling and migration is not considered in their approach.

Likewise, Tighe et al. [66,67] developed a rule-based approach that combines the auto-scaling of applications with dynamic VM allocation to match current workload demands and maintain SLA achievement. In their approach, vertical scaling is performed to scale up and down the VMs according to their resource requirements to run applications, as well as the VMs are consolidated into a minimal number of PMs using live migrations in order to switch off the idle PMs and saving energy costs. As shown on their simulation results, they argued that their combined approach can achieve better application performance with a reduction in VM live migrations compared to the independent approaches. However, their approach only considers the vertical scaling of the scaled resources and do not consider the prediction of these resources. In addition, the costs of the scaled resources are not considered.

Dawoud et al. [68] proposed a dynamic resource provisioning approach that aims to allocate the minimum resources required to handle the future workload demands while maintaining the Service Level Objectives (SLOs). Their approach includes three controllers for CPU, memory, and application to guarantee efficient resource allocation and optimize the application performance. A linear prediction model is used to predict the future resource requirements for efficient allocation and correspond with the workload demands. They have evaluated the proposed approach using the Xen hypervisor with a synthetic workload, and the results show that their controllers are capable to horizontally scale the VMs to correspond with the workload demands while mitigating the SLO violation. However, their approach only considers the horizontal scaling to cope with VMs workload demands without considering the vertical scaling technique. Also, the energy consumption of provisioned resources is not considered.

Moreover, Meng et al. [69] proposed a joint-VM provisioning approach that estimates the VMs capacity needs through statistical multiplexing principles based on their workload patterns. The main idea of this approach is to borrow unused resources from low utilized VMs and reallocated these resources to the VMs with high utilization in order to achieve the application performance requirements. The proposed approach is evaluated based on data collected from commercial data centers using simulations. The results demonstrate that the proposed joint-VM provisioning approach has improved the overall resource utilization by 45% compared to the individual-VM provisioning approaches.

Also, Gandhi et al. [12] investigated the impact of resource auto-scaling on cost, performance and provisioning times for cloud applications. They employed the Amdahl's Law formula to model service time scaling, the queueing-theoretic concepts to model performance scaling, and a Kalman filtering approach to estimate the performance model parameters. They implemented their approach on OpenStack and the results show the ability of the proposed approach to determining the most cost-effective scaling option for a given workload, considering both horizontal and vertical scaling. However, this approach does not consider the prediction of resource requirements and their energy consumption when performing the scaling decisions.

Dutta et al. [8] presented an automatic scaling framework called (SmartScale), which uses a combination of horizontal and vertical scaling in order to optimize the resource usage and the reconfiguration cost incurred due to scaling. The SmartScale is a proactive technique that used a polynomial regression in order to estimate the resource requirements to perform the scaling decisions for the next time interval. They evaluated their framework using a real cloud testbed and the results show that the SmartScale can scale the required resources to run applications with the lowest reconfiguration cost. However, this framework does not consider the power consumption of required resources incurred due to scaling decisions.

5 Overall Discussion

Cloud resource management has the ability to adapt VMs' consolidation and resource provisioning in order to meet the performance requirements of applications, minimize the operation costs and energy consumptions in cloud data centers.

Section 4 has reviewed the related work on VMs' consolidation and resource provisioning mechanisms in cloud environments.

In terms of VMs consolidation, a commonly known NP-hard optimization problem is closely related to it, where the most important objectives are minimizing resource usage and energy consumption, while satisfying the SLAs. As discussed in Section 4.1, the work in Ye et al. [45,49,52] aimed to improve the VMs performance during the migration process, considering the application-aware resource consumption characteristic, but their models only focused on the resource consumption and do not consider the energy consumption overhead of VMs migrations. Moreover, the work presented in Farahnakian et al. [42,43,53–56] mainly focused on dynamic re-allocation of VMs using live migration to increase the overall servers' utilization and minimize the energy consumption, while maintaining the required QoS. Yet, these approaches focused on minimizing PMs energy consumption without taking into consideration the energy consumption incurred by VMs consolidation. Also, the work presented in Verma et al. [58,59] have addressed the issue with migration cost, considering the energy consumption at both PMs and VMs levels. Though there are still limited as the model in Verma et al. [58] does not provide any information regarding the migration cost calculation, whereas, the work in Zakarya et al. [59] is only applicable to the hosts that follow a linear power model and does not consider the heterogeneity of PMs or VMs. Further, the work presented in Farahnakian et al. [46,51,57] employed workload prediction models based on historical data to eliminate unnecessary VM migrations, minimize energy consumption and SLA violations. These models focused on improving the performance of cloud applications by reducing the number of overloaded hosts, but without explicitly considering energy and cost of VMs migrations, as a part of VMs consolidation decision criterion.

In terms of VMs resource provisioning, a fine-grained resource provisioning while ensuring the performance and the SLAs for applications are required, which makes finding the optimal and efficient scaling option a very challenging problem. In Section 4.2, the work in Gandhi et al. [12] investigated the impact of resource auto-scaling on cost, performance, and provisioning times in order to determine the most cost-effective scaling option for cloud applications. Further, the work presented in Ficco et al. [9,66,67] combined the auto-scaling of applications with dynamic VM allocation to match current workload demands and maintain SLA achievement. However, the energy consumption related to the auto-scaling and migration decisions is not considered in their approaches. Moreover, the work presented in Dawoud et al. [8,68,69] considered the prediction of resources provisioning to handle the future workload demand while maintaining the SLOs, but these approaches do not consider the power consumption of required resources incurred due to scaling decisions.

Thus, there is still a need for predictive modelling that dynamically supports VMs live migration and auto-scaling decisions, considering the trade-off between cost, power consumption, and performance

during service operation, which can help cloud providers to make better use of their infrastructures and efficiently manage cloud resources [70,71].

The following Tab. 2 provides a comparison summary of the closely related works on VMs' consolidation and resource provisioning that considers the workload, energy consumption and cost in cloud environments, followed by a comparison summary of the closely related works on the prediction of these mechanisms, as shown in Tab. 3.

Table 2: Summary of existing models for VMs' consolidation and resource provisioning

Criteria by	Workload consideration		Energy consumption consideration		Cost consideration	
	PMs level	VMs level	PMs level	VMs level	Cost of migration	Cost of scaling
[45]	Homogeneous PMs only	Homogeneous VMs only	Not considered	Not considered	Not considered	__
[49]	Heterogeneous PMs	Heterogeneous VMs	Not considered	Not considered	Not considered	__
[52]	Homogeneous PMs only	Heterogeneous VMs	Homogeneous PMs only	Not considered	Not considered	__
[53,54,56]	Heterogeneous PMs	Not considered	Heterogeneous PMs	Not considered	Not considered	__
[42]	Heterogeneous PMs	Heterogeneous VMs	Heterogeneous PMs	Not considered	Not considered	__
[43,55]	Heterogeneous PMs	Not considered	Heterogeneous PMs	Not considered	Considered	__
[59]	Homogeneous PMs only	Homogeneous VMs only	Homogeneous PMs only	Homogeneous VMs only	Considered	__
[58]	Heterogeneous PMs	Heterogeneous VMs	Heterogeneous PMs	Heterogeneous VMs	Considered	__
[9]	Homogeneous PMs only	Not considered	Not considered	Not considered	Considered	Considered
[66,67]	Homogeneous PMs only	Homogeneous VMs only	Homogeneous PMs only	Not considered	Not considered	__
[12]	Homogeneous PMs only	Homogeneous VMs only	Not considered	Not considered	__	Considered

Table 3: Summary of prediction models for VMs' consolidation and resource provisioning

Criteria by	Workload prediction consideration		Energy prediction consideration		Cost estimation consideration	
	PMs level	VMs level	PMs level	VMs level	Cost of migration	Cost of scaling
[46]	Heterogeneous PMs	Heterogeneous VMs	Not considered	Not considered	Not considered	__
[51]	Homogeneous PMs only.	Not considered	Not considered	Not considered	Not considered	__

Table 3 (continued).

Criteria by	Workload prediction consideration		Energy prediction consideration		Cost estimation consideration	
	PMs level	VMs level	PMs level	VMs level	Cost of migration	Cost of scaling
[57]	Heterogeneous PMs	Not considered	Heterogeneous PMs	Not considered	Considered	__
[68]	Homogeneous PMs only	Homogeneous VMs only	Not considered	Not considered	__	Not considered
[69]	Homogeneous PMs only	Homogeneous VMs only	Not considered	Not considered	Not considered	__
[8]	Homogeneous PMs only	Homogeneous and heterogeneous VMs	Not considered	Not considered	__	Considered (horizontal and vertical scaling).

6 Conclusion

This paper has introduced a comprehensive review on the subject of dynamic resource management in cloud computing environments. Firstly, it has introduced the fundamental aspects of cloud computing including its definition, services types, deployment types and virtualization technologies. Secondly, it has presented the concepts of cloud applications and their workload patterns as well as related benchmarks. This is followed by positioning the work in the relevant literature, focusing on cloud resource management issues. A thorough review of related works that focus on VMs consolidation and resource provisioning as well as their predictive technologies has presented. This paper has finally concluded with an overall discussion served as potential research directions, along with a comparison summary of the closely related works.

Funding Statement: The author received no specific funding for this study.

Conflicts of Interest: The author declare that they have no conflicts of interest to report regarding the present study.

References

[1] T. Mukherjee, K. Dasgupta, G. Jung and H. Lee, "An economic model for green cloud," in *Proc. of the 10th Int. Workshop on Middleware for Grids, Clouds and e-Science*, Montreal, Canada, pp. 1–6, 2012.

[2] X. Zhang, J. Lu and X. Qin, "BFEPM: Best fit energy prediction modeling based on CPU utilization," in *2013 IEEE Eighth Int. Conf. on Networking, Architecture and Storage*, Xi'an, China, pp. 41–49, 2013.

[3] J. Conejero, O. Rana, P. Burnap, J. Morgan, B. Caminero *et al.,* "Analyzing hadoop power consumption and impact on application QoS," *Future Generation Computer Systems*, vol. 55, pp. 213–223, 2016.

[4] M. Bagein, J. Barbosa, V. Blanco, I. Brandic, S. Cremer *et al.,* "Energy efficiency for ultrascale systems: Challenges and trends from nesus project," *Supercomputing Frontiers and Innovations*, vol. 2, no. 2, pp. 105–131, 2015.

[5] A. Beloglazov, Y. C. Lee and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Advances in Computers*, vol. 82, pp. 47–111, 2011.

[6] M. Altarawneh, A. Alqaisi and J. B. Salamah, "Evaluation of cloud computing platform for image processing algorithms," *Journal of Engineering Science and Technology*, vol. 14, no. 4, pp. 2345–2358, 2019.

[7] G. Jung, M. A. Hiltunen, K. R. Joshi, R. D. Schlichting and C. Pu, "Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures," in *2010 IEEE 30th Int. Conf. on Distributed Computing Systems*, IEEE, Genova, Italy, pp. 62–73, 2010.

[8] S. Dutta, S. Gera, A. Verma and B. Viswanathan, "SmartScale: Automatic application scaling in enterprise clouds," in *2012 IEEE Fifth Int. Conf. on Cloud Computing*, IEEE, Honolulu, USA, pp. 221–228, 2012.

[9] M. Ficco, C. Esposito, F. Palmieri and A. Castiglione, "A coral-reefs and Game Theory-based approach for optimizing elastic cloud resource allocation," *Future Generation Computer Systems*, vol. 78, pp. 343–352, 2018.

[10] M. Mao and M. Humphrey, "Auto-scaling to minimize cost and meet application deadlines in cloud workflows," in *SC'11: Proc. of 2011 Int. Conf. for High Performance Computing, Networking, Storage and Analysis*, IEEE, Seatle, USA, pp. 1–12, 2011.

[11] Y. Jiang, C. Perng, T. Li and R. Chang, "ASAP: A self-adaptive prediction system for instant cloud resource demand provisioning," in *2011 IEEE 11th Int. Conf. on Data Mining*, IEEE, Vancouver, Canada, pp. 1104–1109, 2011.

[12] A. Gandhi, P. Dube, A. Karve, A. Kochut and L. Zhang, "Modeling the impact of workload on cloud resource scaling," in *2014 IEEE 26th Int. Sym. on Computer Architecture and High Performance Computing*, Jussieu, France, pp. 310–317, 2014.

[13] Amazon, "Amazon EC2 service level agreement," 2017. [Online]. Available: https://aws.amazon.com/ec2/sla/.

[14] P. Berndt and A. Maier, "Towards sustainable IaaS pricing," in *Int. Conf. on Grid Economics and Business Models*, Zaragoza, Spain, pp. 173–184, 2013.

[15] M. Al-Roomi, S. Al-Ebrahim, S. Buqrais and I. Ahmad, "Cloud computing pricing models: A survey," *International Journal of Grid and Distributed Computing*, vol. 6, no. 5, pp. 93–106, 2013.

[16] Q. Zhang, L. Cheng and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.

[17] P. M. Mell and T. Grance, "The NIST definition of cloud computing," *Recommendations of the National Institute of Standards and Technology*, Gaithersburg, MD, 2011.

[18] S. Nanda and T. C. Chiueh, "A survey on virtualization technologies," Rpe Report, 2005.

[19] Y. Li, Y. Wang, B. Yin and L. Guan, "An online power metering model for cloud environment," in *2012 IEEE 11th Int. Sym. on Network Computing and Applications*, Cambridge, USA, pp. 175–180, 2012.

[20] OpenNebula, "The simplest cloud management experience." 2018. [Online]. Available: https://opennebula.org/.

[21] OpenStack, "Open source software for creating private and public clouds." 2019. [Online]. Available: https://www.openstack.org/.

[22] CloudStack, "Apache CloudStackTM open source cloud computing." 2019. [Online]. Available: http://cloudstack.apache.org/.

[23] A. Vogel, D. Griebler, C. A. F. Maron, C. Schepke and L. G. Fernandes, "Private IaaS clouds: A comparative analysis of OpenNebula, CloudStack and OpenStack," in *2016 24th Euromicro Int. Conf. on Parallel, Distributed, and Network-Based Processing (PDP)*, Heraklion, Greece, pp. 672–679, 2016.

[24] OpenQRM, "Professional open-source data center and cloud management." 2019. [Online]. Available: https://openqrm-enterprise.com/.

[25] EUCALYPTUS, "Eucalyptus." 2019. [Online]. Available: https://www.eucalyptus.cloud/.

[26] Nimbus, "Nimbus is cloud computing for science." 2019. [Online]. Available: http://www.nimbusproject.org/.

[27] R. Dua, A. R. Raja and D. Kakadia, "Virtualization vs containerization to support PaaS," in *2014 IEEE Int. Conf. on Cloud Engineering*, Boston, USA, pp. 610–614, 2014.

[28] KVM, "Kernel-based virtual machine." 2018. [Online]. Available: https://www.linux-kvm.org/.

[29] Xen, "Xen project." 2019. [Online]. Available: https://xenproject.org/.

[30] VMware, "VMware cloud." 2019. [Online]. Available: https://www.vmware.com/.

[31] Oracle, "Virtual box." 2019. [Online]. Available: http://www.virtualbox.org/.

[32] Cisco, "Linux containers : Why they're in your future and what has to happen first." 2019. [Online]. Available: https://www.redhat.com/en/blog/linux-containers-why-theyre-your-future-and-what-has-happen-first.

[33] Docker, "Enterprise container platform for high-velocity innovation." 2019. [Online]. Available: https://www.docker.com/.

[34] B. Business, "How to jump from cloud to cloud." 2019. [Online]. Available: https://www.linux.com/news/how jump-cloud-cloud/.

[35] Linux, "LXC linux containers." 2019. [Online]. Available: https://linuxcontainers.org/.

[36] CloudFoundry, "Cloudfoundry warden manages isolated, ephemeral, and resource controlled environments." 2019. [Online]. Available: https://github.com/cloudfoundry-attic/warden.

[37] C. Fehling, F. Leymann, R. Retter, W. Schupeck and P. Arbitter, *Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications*, Springer, 2014.

[38] M. Ficco, M. Rak, S. Venticinque, L. Tasquier and G. Aversano, "Cloud evaluation: Benchmarking and monitoring," in *Quantitative Assessments of Distributed Systems*, John Wiley & Sons, Hoboken, USA, pp. 175–199, 2015.

[39] SPEC, "Benchmark overview-SPEC cloud IaaS 2018 benchmark." 2019. [Online]. Available: https://www.spec.org/cloud_iaas2018/docs/faq.html.

[40] SPEC, "SPEC-Standard Performance Evaluation Corporation." 2019. [Online]. Available: https://www.spec.org/.

[41] Stress-ng, "Stress tests." 2018. [Online]. Available: http://kernel.ubuntu.com/~cking/stress-ng/.

[42] F. Farahnakian, R. Bahsoon, P. Liljeberg and T. Pahikkala, "Self-adaptive resource management system in IaaS clouds," in *2016 IEEE 9th Int. Conf. on Cloud Computing (CLOUD)*, San Francisco, USA, pp. 553–560, 2016.

[43] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.

[44] O. Alrajeh, M. Forshaw and N. Thomas, "Machine learning models for predicting timely virtual machine live migration," in *European Workshop on Performance Engineering*, Berlin, Germany, pp. 169–183, 2017.

[45] K. Ye, X. Jiang, D. Huang, J. Chen and B. Wang, "Live migration of multiple virtual machines with resource reservation in cloud computing environments," in *2011 IEEE 4th Int. Conf. on Cloud Computing*, Washington, DC, USA, pp. 267–274, 2011.

[46] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila and H. Tenhunen, "Utilization prediction aware VM consolidation approach for green cloud computing," in *2015 IEEE 8th Int. Conf. on Cloud Computing*, New York, USA, pp. 381–388, 2015.

[47] M. R. Hines and K. Gopalan, "Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning," in *Proc. of the 2009 ACM SIGPLAN/SIGOPS Int. Conf. on Virtual execution environments*, Washington, DC, USA, pp. 51–60, 2009.

[48] B. R. Raghunath and B. Annappa, "Virtual machine migration triggering using application workload prediction," *Procedia Computer Science*, vol. 54, pp. 167–176, 2015.

[49] H. Zhao, Q. Zheng, W. Zhang, Y. Chen and Y. Huang, "Virtual machine placement based on the VM performance models in cloud," in *2015 IEEE 34th Int. Performance Computing and Communications Conf. (IPCCC)*, Nanjing, China, pp. 1–8, 2015.

[50] T. C. Ferreto, M. A. S. Netto, R. N. Calheiros and C. A. F. De Rose, "Server consolidation with migration control for virtualized data centers," *Future Generation Computer Systems*, vol. 27, no. 8, pp. 1027–1034, 2011.

[51] A. Beloglazov and R. Buyya, "Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1366–1379, 2013.

[52] F. Xu, F. Liu, L. Liu, H. Jin, B. Li *et al.,* "iAware: Making live migration of virtual machines interference-aware in the cloud," *IEEE Transactions on Computers*, vol. 63, no. 12, pp. 3012–3025, 2014.

[53] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *2010 10th IEEE/ACM Int. Conf. on Cluster, Cloud and Grid Computing*, Melbourne, Australia, pp. 826–831, 2010.

[54] A. Beloglazov, J. Abawajy and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.

[55] A. Beloglazov and R. Buyya, "Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers," in *Proc. of the 8th International Workshop on Middleware for Grids, Clouds and e-Science*, Bangalore, India, pp. 1–6, 2010.

[56] M. H. Malekloo, N. Kara and M. El Barachi, "An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments," *Sustainable Computing: Informatics and Systems*, vol. 17, pp. 9–24, 2018.

[57] H. Zhou, Q. Li, K. K. R. Choo and H. Zhu, "DADTA: A novel adaptive strategy for energy and performance efficient virtual machine consolidation," *Journal of Parallel and Distributed Computing*, vol. 121, pp. 15–26, 2018.

[58] A. Verma, P. Ahuja and A. Neogi, "pMapper: Power and migration cost aware application placement in virtualized systems," in *ACM/IFIP/USENIX Int. Conf. on Distributed Systems Platforms and Open Distributed Processing*, Leuven, Belgium, pp. 243–264, 2008.

[59] M. Zakarya and L. Gillam, "An energy aware cost recovery approach for virtual machine migration," in *Int. Conf. on Economics of Grids, Clouds, Systems and Services*, Athens, Greece, pp. 175–190, 2016.

[60] Amazon Web Services, "AWS." 2019. [Online]. Available: https://aws.amazon.com/.

[61] J. Yang, C. Liu, Y. Shang, B. Cheng, Z. Mao *et al.,* "A cost-aware auto-scaling approach using the workload prediction in service clouds," *Information Systems Frontiers*, vol. 16, no. 1, pp. 7–18, 2014.

[62] A. Y. Nikravesh, S. A. Ajila and C. H. Lung, "An autonomic prediction suite for cloud resource provisioning," *Journal of Cloud Computing*, vol. 6, no. 1, pp. 3, 2017.

[63] J. Yang, C. Liu, Y. Shang, Z. Mao and J. Chen, "Workload predicting-based automatic scaling in service clouds," in *2013 IEEE Sixth Int. Conf. on Cloud Computing*, Santa Clara, USA, pp. 810–815, 2013.

[64] Q. Zhang, H. Chen and Z. Yin, "PRMRAP: A proactive virtual resource management framework in cloud," in *2017 IEEE Int. Conf. on Edge Computing (EDGE)*, Honolulu, USA, pp. 120–127, 2017.

[65] F. Lombardi, A. Muti, L. Aniello, R. Baldoni, S. Bonomi *et al.,* "PASCAL: An architecture for proactive auto-scaling of distributed services," *Future Generation Computer Systems*, vol. 98, pp. 342–361, 2019.

[66] M. Tighe and M. Bauer, "Integrating cloud application autoscaling with dynamic VM allocation," in *2014 IEEE Network Operations and Management Sym. (NOMS)*, Krakow, Poland, pp. 1–9, 2014.

[67] M. Tighe and M. Bauer, "Topology and application aware dynamic VM management in the cloud," *Journal of Grid Computing*, vol. 15, no. 2, pp. 273–294, 2017.

[68] W. Dawoud, I. Takouna and C. Meinel, "Elastic VM for cloud resources provisioning optimization," in *Int. Conf. on Advances in Computing and Communications*, Kochi, India, pp. 431–445, 2011.

[69] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet *et al.,* "Efficient resource provisioning in compute clouds via VM multiplexing," in *Proc. of the 7th Int. Conf. on Autonomic computing*, Washington, DC, USA, pp. 11–20, 2010.

[70] M. Aldossary and K. Djemame, "Performance and energy-based cost prediction of virtual machines live migration in clouds," in *Proc. of the 8th Int. Conf. on Cloud Computing and Services Science (CLOSER)*, Madeira, Portugal, pp. 384–391, 2018.

[71] M. Aldossary and K. Djemame, "Performance and energy-based cost prediction of virtual machines auto-scaling in clouds," in *2018 44th Euromicro Conf. on Software Engineering and Advanced Applications (SEAA)*, Prague, Czech Republic, pp. 502–509, 2018.